Technical Article
Using PMICs to Support a Wide Range of Power Sequencing Requirements

Karl Wallinger

Nov 1, 2016

When designing a power solution for an application processor, your first considerations are often the number of rails needed, the output voltage and the maximum load current. There are tools like the Quick Search that can help you select the right power-management IC (PMIC) for the processor or application, whether it's an industrial application like Factory Automation or Human Machine Interface (HMI), or an automotive application like infotainment or advanced drivers assistance systems (ADAS).

But you also need to ask yourself:

• How will you handle power-up and power-down sequencing?
• How will you ensure the completion of the power-down sequence when power is removed?

Let's take a look at these questions in more detail.

Power-up and Power-down Sequencing

Power-up and power-down sequencing are very important and are specified in the processor's data manual. If the PMIC has enable pins and voltage-selection pins for each rail, one idea for handling power-up and power-down sequencing would be to have each PMIC output enable the next regulator in the sequence. This has a few drawbacks, however, since the output voltage needs to be high enough to send a logic-high signal to the enable, and it also makes a reverse power-down sequence very difficult, if not impossible.

A second approach is to use a microcontroller's general-purpose input/output (GPIO) pins to enable each rail in sequence to meet the sequencing requirements. However, this requires extra pins (and therefore a bigger package), as well as a sequencing microcontroller and some firmware to get the system running.

An easier way is to use the PMIC's one-time programmable (OTP) memory, which contains default output voltages and power-up and power-down sequencing for the device. So when the PMIC is enabled or disabled, a pre-programmed sequence will execute without any interaction from a microcontroller. The PMIC can also start the power sequence within a few milliseconds, which leads to faster boot times than when booting up a microcontroller before running power sequencing.

Using the PMIC in different configurations doesn't require any different firmware, or actually any firmware at all. For example, the TPS659037 has two different configurations, based on the orderable part number, to power the AM572x Sitara™ depending on processor frequency and the number of cores used. The PMIC will enable or disable core rails depending on the configuration. So you can use the same PMIC in two different configurations, with no hardware changes or additional firmware development. Configurations for other processors or applications are possible by programming a different sequence in the OTP memory.
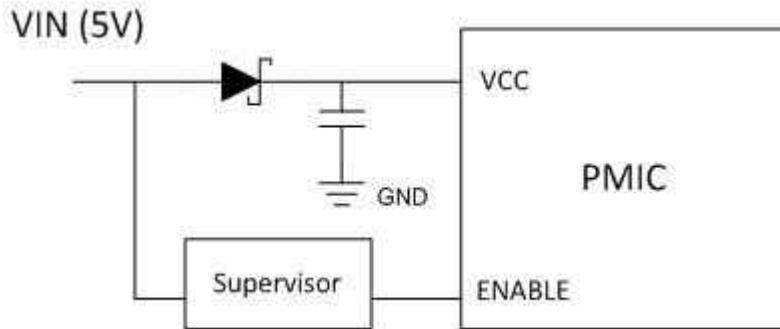
Power-down Sequence When Power Is Removed Unexpectedly

There is often more to the power-down sequence than just meeting the timing requirements. Even though the PMIC contains a power-down sequence that meets the processor's timing requirements, what happens when power is removed unexpectedly? If the input voltage to the PMIC is removed very quickly, there won't be enough input power to maintain the output voltage during sequencing. In a simple solution, all rails will discharge based on output capacitance and load current. It may not be easy to predict the ramp-down rates, so it's possible that the power-off could happen out of order. So how do you fix this?
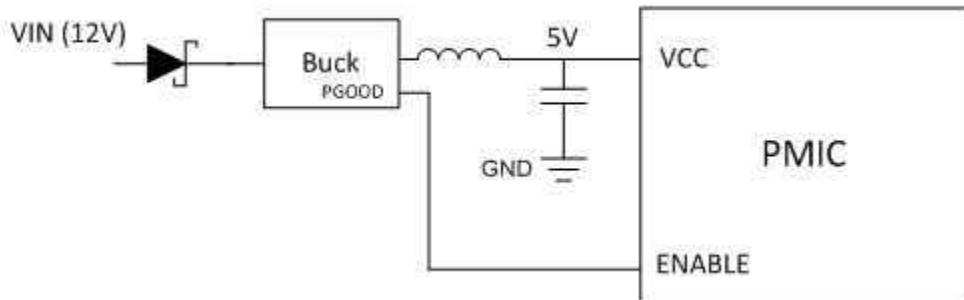
You need a way to block reverse current, a method to store energy and a disable signal to the PMIC. A Schottky diode can block reverse current when the input is removed. Capacitors can hold up the input voltage while the power-down sequence occurs. Disabling the PMIC will depend on the system configuration, however. In many cases, it is desirable to have an always-on system, so let's consider a case in which the system will be enabled when power is first supplied. You can create an always-on system in one of two ways:

- A supervisor, which creates a logic signal to indicate that power is good, for systems where V_{IN} directly powers the PMIC.
- A power-good signal for systems using a pre-regulator to generate the PMIC supply.

Figure 1 shows the implementation of the first option, while Figure 2 shows the implementation of the second option.

Figure 1. Supporting Uncontrolled Power Down When v_{IN} Powers the PMIC

Figure 2. Supporting Uncontrolled Power Down When a Pre-regulator Powers the PMIC

In the case of the pre-regulator, the pre-regulator output capacitance can also act as the energy storage to hold VCC up. You should base your chosen total supply capacitance on the worst-case leakage current during power down so that the voltage is held up until the power-down sequence completes.

Use Equation 1 to calculate the required capacitance:

$$C = I * \frac{\Delta T}{Vcc - Vmin} \qquad (1)$$

Where I is the leakage current, Vcc is the supply voltage to the PMIC, Vmin is the minimum input voltage the PMIC needs to operate, and ΔT is the time it takes the power-down sequence to complete. For TI's TPS659037 and TPS65916, Vmin is 2.75V, and the pre-programmed power-down sequence is typically 1ms.

See our PMIC page to learn about more ways to enable your system power with TI's broad portfolio of scalable PMICs.

Additional Resources

- Download these user's guides:
  - "TPS659037 User's Guide to Power AM572x and AM571x."
  - "TPS65916 User's Guide to Power AM571x."
  - Check out the AM572x evaluation module.

IMPORTANT NOTICE AND DISCLAIMER